# Camera movement

**Keyboard handling**

Whenever one of the special keys (w,a,s,d, arrows, …) is pressed, the code of the key is saved in an object, so also multiple keys can be pressed at one time.

For back and forth (key W and S) the speed variable “backAndForth” is set to a positive or negative value. Each rendering cycle the elapsed time is multiplied with this value, which is used to calculate the new (x,y,z)-value with which the viewMatrix is translated with.

The same principle is used for left, right and up, down. A variable is set to a positive or negative value and each rendering cycle, a new value is calculated with which the viewMatrix is rotated.

**Mouse handling**

At mouse-button-down the position is saved. Then when moving the mouse, every difference (delta value) is calculated and is added to pitch and yaw (our variables for rotating the viewMatrix) until the mouse-button is released. (mouse-button-up event)  
  
**Animated camera flight**

For our animated camera flight we save the important data (duration, newPosX, newPosY, newPosZ newPitch, newYaw, stay) in a queue and whenever one queue element is executed, the next one is loaded and executed.

This algorithm is pretty simple. The first time from the animated camera flight, we need to load the first queue element and calculate the delta values of the new position (where the camera should move). These delta values are calculated by subtracting the current position from the new position and dividing it by the duration. When this is done, in every render cycle the delta is multiplied by the elapsed time to get the new interpolated position.